REMARKS

Claims 1-15 are active in the application. Claims 1, 3, 5, and 12 have been amended to improve clarity. No new matter has been added and the scope of the claims has not been changed.

Claims 1-12 and 15 were rejected under 35 USC 102 (e) as being anticipated by US patent 6,301,709 to Warmink. Claims 13 and 14 were rejected under 35 USC 103 (a) as being unpatentable over Warmink in view of US patent 5,884,091 to Ghori et al. These rejections are traversed.

The present invention provides a system for updating software in an electronic device having two processors. The processors may be in different locations. In the present system, a second processor monitors a first processor for fatal bugs or software errors. If a fault is detected in the first processor, the faulty first processor can be reset by the second processor. After reset, error-free software can be loaded into the first processor.

In the present invention, monitoring is preferably accomplished by transmitting an inquiry signal from the second processor to the first processor. Inquiry signals are sent at predetermined time intervals. When operating normally, the first processor replies to each inquiry signal. Reception of the response signal indicates to the second processor that a fault has not occurred. If a fault occurs in the first processor, the first processor is rendered unable to respond, and no response is received by the second processor. When a fault is detected in this way, the second processor transmits a compulsory reset signal to the first processor, causing it to reset. Thereafter, new, error-free software is sent to the first processor. In this way, the present invention allows repair and correction of corrupted software in the first processor, even if the first processor cannot communicate. These aspects of the present invention are discussed in detail at page 13, line 20 through page 14, line 17, and at page 17, line 25 through page 18, line 12.

The present invention avoids problems caused by corrupted software in conventional systems. For example, page 2, lines 21-25 describe problems in conventional systems where corrupted software (firmware) prevents the first processor from communicating. In this case, replacement of the software must be done manually, or

must be done with hardware intervention. In conventional systems, there is no way to update software in a processor that cannot communicate due to a software error.

These features are expressed in claim 1, which requires that the second processor "executes an update control of said program when a fault of said first processor is detected". These features are also expressed in claim 12, which requires the step of "transmitting a compulsory reset signal from said second processor to said first processor to stop an operation of said first processor when said response pulse can not be detected within a predetermined period". In the present invention, the first processor is determined to be faulty when it fails to respond to inquiry signals, or responds improperly.

Warmink teaches a system with software upgrade capability that has the same problems as conventional systems described in the present specification. Warmink does not teach or suggest fault detection, and does not teach or suggest the capability (e.g., a compulsory reset function) to upgrade software in a processor that fails to communicate. Warmink teaches a system that compares software versions, and determines which version is desired or most recent. Newer software is transmitted to circuits that have older or outdated software. For example, Warmink described "version information" that is analyzed to determine whether software should be replaced (see col. 2, lines 1-28, col. 2, lines 40-50). Warmink specifically recommends that version information be used to determine which software is oldest, and recommends that the older software be overwritten with newer software. For example, Warmink states in col. 2, lines 47-50: "For example, the determination may be whether the data in the master circuit pack is a newer version of software or other information that the slave circuit pack should have." Warmink also states in col. 2, lines 59-63: "...upon receipt by either or both circuit packs of a system signal, the two circuit packs exchange version information, and whichever of the two circuit packs has older or less preferred data then receives the newer data from the other circuit pack. " Using version information to determine which circuit pack has the oldest software is also described in col. 3, lines 1-25, and col. 4, line 57 through col. 5, line 3 of Warmink.

It is important to note that, in Warmink, in order for a circuit to receive updated software, it must be able to communicate the version information. This is very different from the present invention in which updated software is sent only if a fault (e.g. lack of

communication) is detected. Warmink does not teach or suggest fault detection, and does not teach or suggest updating software or a compulsory reset command in response to detection of a circuit fault or malfunction (e.g. failure to send a response signal). Warmink is only concerned with updating old software versions with newer or more preferred versions. Warmink offers absolutely no teaching as to how to correct a circuit fault or software bug. Accordingly, Warmink fails to meet the limitations of claims 1 or 12, and the rejections of these claims must therefore be withdrawn.

Regarding claim 2, Warmink does not teach or suggest that a circuit pack monitors a response pulse, and transmits a compulsory reset signal when the response pulse cannot be detected. The Office Action argues that col. 3, 30-55 and col. 2, 65-67 teach the resetting function and updating when a pulse is not detected. This is erroneous. Col. 3 30-55 actually teaches that some circuit packs can be programmed to not receive new software, since "an older version of data may be beneficial" (col. 3, lines37-38). Col. 2, lines 65-67 actually teaches that a circuit pack does not need to receive an update initiation signal to participate in software upgrading. This section says nothing about steps to take if a circuit pack cannot communicate.

Regarding claims 3 or 5, Warmink does not teach or suggest an activating pulse that causes a circuit pack or processor to transmit a reset signal.

Regarding claims 8 or 10, Warmink does not teach or suggest transmitting a compulsory reset signal when an activation response pulse can not be detected. Warmink does not teach or suggest any action when a processor or circuit pack fails to communicate.

Regarding claims 14 and 15, Warmink does not teach or suggest executing a stop control of a circuit pack or processor when a response pulse can not be detected within a predetermined time.

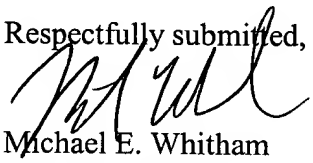
The Office relies upon Ghori et al. as teaching the step of disabling a processor during a reset operation. Ghori et al. do not teach or suggest fault detection, or detection of a failure of a processor to transmit response signals. Ghori et al. also does not teach or suggest updating processor software when a fault is detected. Hence, no conceivable combination of Ghori et al. with Warmink can produce the present invention as claimed.

In view of the foregoing, it is respectfully requested that the application be reconsidered, that claims 1-15 be allowed, and that the application be passed to issue.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

A provisional petition is hereby made for any extension of time necessary for the continued pendency during the life of this application. Please charge any fees for such provisional petition and any deficiencies in fees and credit any overpayment of fees for the petition or for entry of this amendment to Attorney's Deposit Account No. 50-2041 (Whitham, Curtis & Christofferson P.C.).

Respectfully submitted,



Michael E. Whitham
Reg. No. 32,635

Whitham, Curtis, & Christofferson, P.C.
11491 Sunset Hills Road, Suite 340
Reston, VA, 20190

Phone: 703-787-9400
Fax: 703-787-7557